

Multimedia Systems

Lecture – 8

By

Dr. Priyambada Subudhi
Assistant Professor
IIIT Sri City

Assessment Plan

Sl. No.	Mode of Assessment	% of Marks
1	Mid Semester	20
2	End Semester	30
3	Scheduled Quiz	20
4	Surprised Quiz	10
5	Project	20

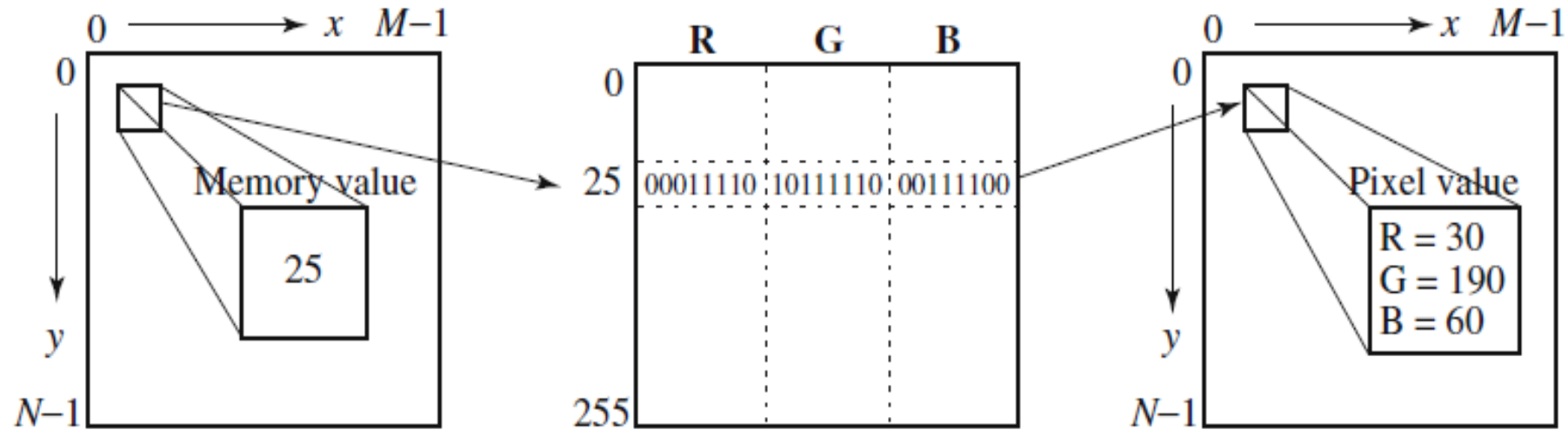
8-bit color images

- Many systems can utilize color information stored with only 8 bits of information (the so-called 256 colors) in producing a screen image.
- Such image files use the concept of a **lookup table** to store color information.
 - Basically, the image stores not color, but instead just a set of bytes, each of which is actually an index into a table with 3-byte values that specify the color for a pixel with that lookup table index.

Color Lookup Tables

- The idea used in 8-bit color images is to store only the index, or code value, for each pixel.
- Then, if a pixel stores, say, the value 25, the meaning is to go to row 25 in a color lookup table (LUT).
- For an 8-bit image, the image file can store in the file header information just what 8-bit values for R, G, and B correspond to each index.
- The LUT is often called a *palette*.

Color LUT for 8-bit color images

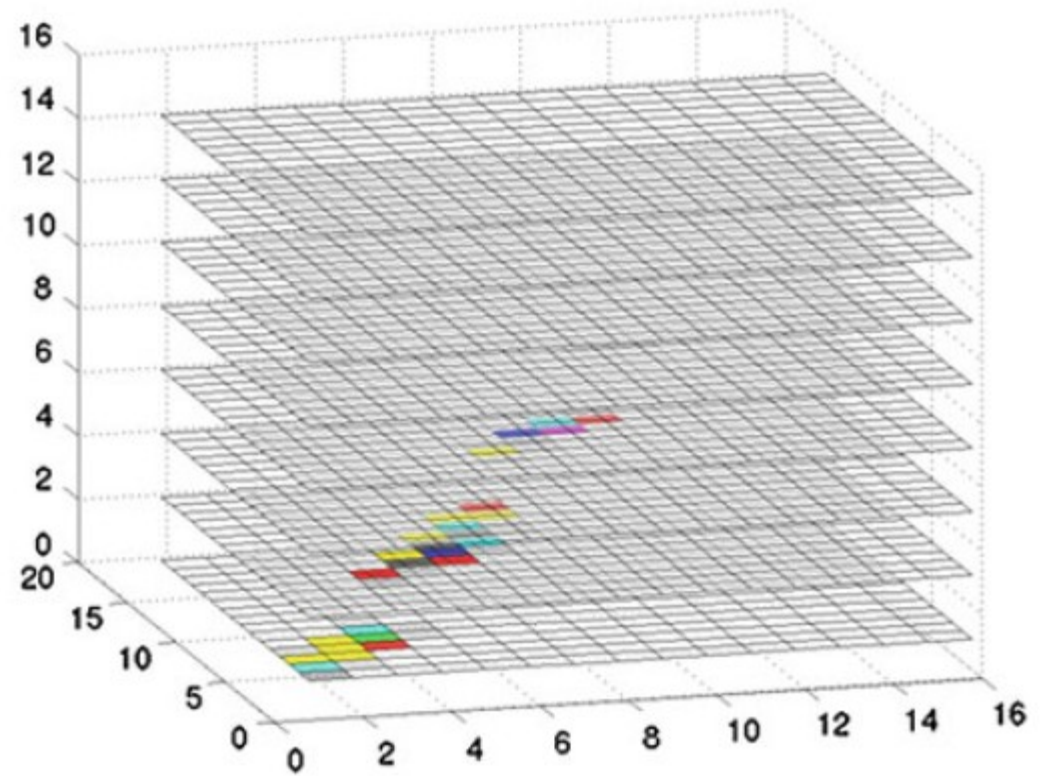


Color Histogram

- It makes sense to carefully choose just which colors to represent best in the image.
 - if an image is mostly red sunset, it is reasonable to represent red with precision and store only a few greens.
- Suppose all the colors in a 24-bit image were collected in a $256 \times 256 \times 256$ set of cells, along with the count of how many pixels belong to each of these colors stored in that cell.
 - For example, if exactly 23 pixels have RGB values (45, 200, 91) then store the value 23 in a three-dimensional array, at the element indexed by the index values [45, 200, 91]
- This data structure is called a *color histogram*.

3D scatterplot of RGB colors in forestfire image

- The histogram has $16 \times 16 \times 16$ bins and shows the count in each bin in terms of intensity and pseudocolor.
- We can see a few important clusters of color information, corresponding to the reds, yellows, greens, and so on, of the forestfire image.
- Basically, large populations in 3D histogram bins can be subjected to a split-andmerge algorithm to determine the "best" 256 colors



Example of an 8-bit color image

Note the great savings in space for 8-bit images over 24-bit ones: a 640×480 8-bit color image requires only 300kB of storage, compared to 921.6kB for a color image.



24-bit color image



8-bit color image

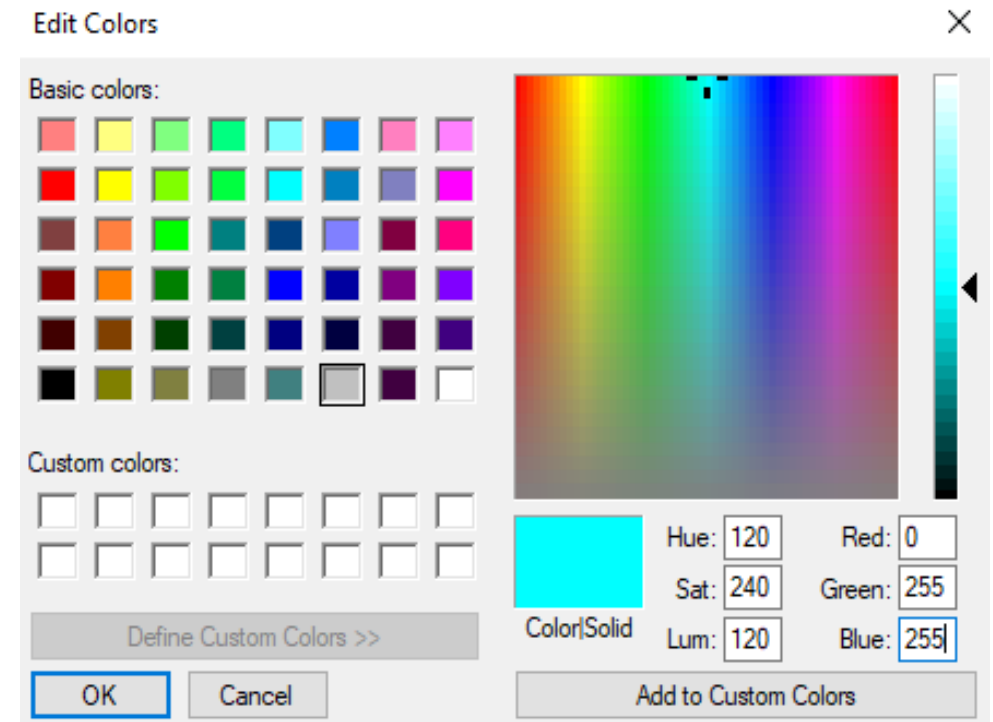
How to Transform to 8 bit RGB

How to Select the Best 256 LUT RGB Entries Without Constructing a Color Histogram

- Transform 3 bytes RGB to 8bit per pixel for 256 selective LUT table index by selecting RGB color separately.
- Sort **R** byte in the original image to select 8 (**3bits**) most popular R intensities.
 - Those 8 most popular R byte codes to LUT.
- Sort **G** byte to select 8 (**3bits**) most popular G intensities
 - Those 8 most popular G byte codes to LUT
- Sort **B** byte to select 4 (**2bits**) most popular B intensities.
 - Those 4 B byte codes to LUT
- All the combinations of **8R x 8G x 4B** becomes 256 LUT RGB entries.
- Transformed **3+3+2** bits per pixel is an index to transformed LUT.
- Each pixel value in the original RGB is transformed to index value of the nearest RGB entries in LUT by comparing the original to RGB LUT entries.

Color Picker

- A *color picker* consists of an array of fairly large blocks of color (or a semicontinuous range of colors) such that a mouse click will select the color indicated.
- In reality, a color picker displays the palette colors associated with index values from 0 to 255.
- If the user selects the color block with index value 2, then the color meant is cyan, with RGB values (0, 255, 255).



Color picker for 8-bit color: each block of the color picker corresponds to one row of the color LUT

												R			G			B			
0 0 0 0	1 1 1 1	2 2 2 2	→	0	0	255	255	Cyan													
0 0 0 0	1 1 1 1	2 2 2 2		1																	
0 0 0 0	1 1 1 1	2 2 2 2		2																	
0 0 0 0	1 1 1 1	2 2 2 2		3																	
0 0 0 0	1 1 1 1	2 2 2 2		4																	
				⋮																	
3 3 3 3	4 4 4 4	5 5 5 5																			
3 3 3 3	4 4 4 4	5 5 5 5																			
3 3 3 3	4 4 4 4	5 5 5 5																			
3 3 3 3	4 4 4 4	5 5 5 5																			
6 6 6 6	7 7 7 7	8 8 8 8																			
6 6 6 6	7 7 7 7	8 8 8 8																			
6 6 6 6	7 7 7 7	8 8 8 8																			
6 6 6 6	7 7 7 7	8 8 8 8																			
															255						